Instrumente si Tehnici de Baza in Informatica

Semestrul I 2024-2025

Vlad Olaru

Curs 10 - outline

- stocarea datelor
  - echipamente de tip bloc
  - sisteme de fisiere

Stocarea datelor

- memoria principala – singurul mediu de stocare de dimensiune mare accesibil direct procesorului
  - acces random
  - uzual volatila
  - tipic Dynamic Random-Access Memory (DRAM)
- stocarea secundara – extensie a memoriei principale care furnizeaza capacitate mare de stocare **nevolatila**
  - discuri dure (hard disks)
  - discuri flexibile (floppy disks)
  - CDROM/DVD
  - SSD (Solid State Disks)
  - flash drives
  - samd

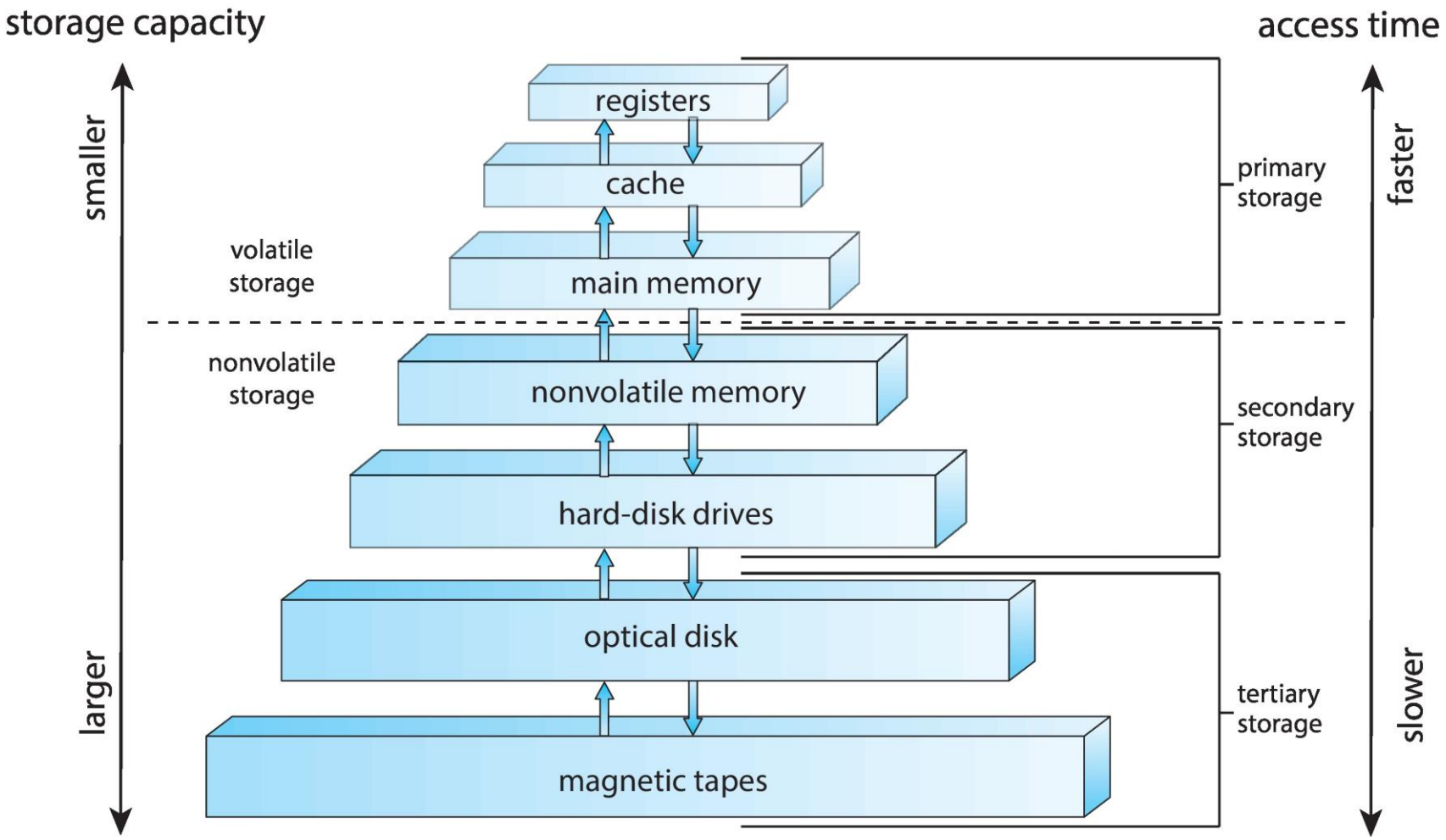
Stocarea datelor (cont.)

- **discuri dure (Hard Disk Drives, HDD**) – platane rigide din metal sau sticla acoperite cu material magnetic capabil de stocarea datelor
  - platanele se invart in jurului unui ax de rotatie
  - suprafata unui platan divizata logic in **piste**
  - pista divizata la randul ei in **sectoare**
  - uzual exista mai multe platane => in 3D colectia de piste egal departate de axul de rotatie formeaza un **cilindru**
  - controller-ul de disc (**disk controller**) determina interactiunea logica dintre echipament si calculator
- **echipamente cu memorie nevolatila (Non-volatile memory, NVM**)– mai rapide decat HDD, nevolatile
  - diverse tehnologii
  - devin din ce in ce mai raspandite

Ierarhia de memorie

- sistemele de stocare organizate in ierarhii in functie de
  - viteza
  - cost
  - volatilitate
- **caching**
  - copiaza informatia in sisteme de stocare mai rapide
  - memoria principala vazuta drept cache pentru stocarea secundara
- **driverul de echipament (Device Driver)**
  - componenta kernel specializata, opereaza fiecare device controller pt a gestiona operatiile de I/O
  - furnizeaza o interfata uniforma intre controller si kernel

Ierarhia echipamentelor de memorie



Echipamente de tip bloc in Unix

- unitatea de transfer a datelor = blocul de date
- arhitecturile de calcul moderne bazate pe Direct Memory Access (DMA)
  - transfer de date intre RAM si echipamentul bloc neintermediat de CPU
  - suport HW sub forma unui chip programabil specializat
  - contrar tehnicii de *programmed I/O*
- reprezentate in sistemele Unix ca fisiere speciale de tip bloc in directorul */dev*

  Ex: */dev/sda1, /dev/cdrom*
- caracterizate de *nr major* si *nr minor*
  - primul identifica driverul asociat echipamentului, al doilea nr unitatii de acel tip din sistem
- create cu *mknod*

  Ex: *$ mknod /dev/sdc1 b 8 1*

Echipamente loop (loop devices)

- *pseudo-echipament (pseudo-device)* care permite folosirea unui fisier obisnuit (*regular file*) ca echipament bloc
- ex: sistem de fisiere continut intr-un singur fisier
  - imagini ISO de CDROM/DVD sau imagini floppy disk
  - se pot instala in sistem cu comanda *mount* fara suport HW
- uzual */dev/loop0, /dev/loop1*, samd.
- *losetup*
  - asociaza device-uri loop cu fisiere obisnuite sau echipamente bloc

```
Ex:     $ losetup -f file.img     # asociaza primul loop device disponibil
                                  # cu fisierul furnizat ca argument
        $ losetup -j file.img     # afiseaza starea loop devices asociate
                                  # cu fisierul imagine argument
```

Utilizarea echipamentelor bloc

- stocarea datelor
  - datele inregistrate pe discuri cf unui format specific unui anumit tip de sistem de fisiere
  - uzual, formatul genereaza structura logica de acces cu fisiere si respectiv colectii de fisiere (directoare)
  - inregistrarea formatului sistemului de fisiere pe disc (“formatarea discului”)
    - se folosesc comenzi speciale pt fiecare tip de sistem de fisiere

      Ex: *$ mkfs –t ext4 /dev/sda1* # ⇔ *$ mkfs.ext4 /dev/sda1*

- spatiu de swap
  - discuri neformatate (in sensul de mai sus) folosite de memoria virtuala a sistemului de operare
  - programele utilizator pot fi mai mari decat memoria RAM disponibila
  - memorie RAM insuficienta pt noi procese
  - creat cu *mkswap,* activat cu *swapon,* dezactivat cu *swapoff*

    Ex: *$ mkswap /dev/sda2 ; swapon /dev/sda2*

Comenzi utile

- *parted/fdisk* - manipuleaza tabela de partitii a unui disc

  $ parted –l /dev/sda # afiseaza tabela de partitii a */dev/sda*

- *blkid/lsblk* – furnizeza informatii/atribute ale echipamentelor bloc

  $ blkid; lsblk –f # atribute discuri + informatii despre FS

- *df* – raporteaza informatii despre utilizarea spatiului de disc

  $ df –h # afisare “human readable”

- *du* – estimeaza utilizarea spatiului folosit de fisiere

  *$ du -sh ** # dim totala a fisierelor din directorul curent

- *free* – afiseaza memoria si spatiul de swap disponibile

  $ free –h # human readable

Fisiere

- fisier
  - abstractie de nivel de sistem de operare pt stocarea persistenta a datelor
  - concret, containere pt stocarea persistenta a datelor
  - la nivelul cel mai de jos, stocarea persistenta se face pe discuri
  - uzual referite prin nume (string ASCII) convertit la o reprezentare interna a kernelului de catre sistemul de fisiere
  - paradigma uzuala de folosire: *open – read/write – close*
- pe langa date, fisierele stocheaza si *metadate:*
  - atribute: data ultimului acces, ultimei modificari, proprietarul fisierului, permisiuni, dimensiunea fisierului, etc
  - structura de acces la reprezentarea low-level a datelor (adrese de blocuri de disc)

Sistemul de fisiere

- componenta speciala a SO care gestioneaza fisierele si directoarele
- gestioneaza mediul de stocare persistenta a datelor
  - structureaza datele pe disc intr-un anumit *format*
- ofera utilizatorului o interfata uniforma de acces la date
  - bazata pe abstractia de fisier si operatiile (apeluri sistem) de lucru cu ele
- SO moderne capabile sa integreze sisteme de fisiere cu format diferit in aceeasi ierarhie de directoare
  - VFS – Virtual Filesystem Switch (ext3, ext4, ntfs, vfat, etc)
- disponibil utilizatorului ca urmare a operatiei de *mount*

  $ mount –t ext4 /dev/sda1 /

  $ umount /dev/sda1 # operatia inversa, merge si *umount /*
- directorul in care se instaleaza discul formatat s.n. *mountpoint*

/etc/fstab

- tabela system-wide cu mountpoint-uri inspectata la bootarea SO
- la bootare, mountpoint-urile din tabela se instaleaza ca si cand s-ar fi apelat

  $ mount -a

- suplimentar, *systemd* instaleaza unit-urile de tip *mount*
- mountpoint-urile active la un moment dat disponibile in */etc/mtab* sau */proc/mounts*

  $ mount # afiseaza informatia din tabele

- fiecare mountpoint si sistemul de fisiere asociat sunt descrise pe o linie din *fstab*
  - fiecare linie contine 6 campuri

Campurile fstab

- (1) *fs_spec* – echipamentul bloc sau sistemul de fisiere remote care trebuie instalat

Ex: */dev/sda7, /dev/cdrom, fmi.unibuc.ro:/home*

  - se pot folosi LABELS sau UUIDs

    – metoda preferabila, numele de device se poate schimba cand se adauga/indeparteaza device-uri

    $ blkid # sau *lsblk -f*

- (2) *fs_file* – mountpoint-ul (sau *none*, pt partitia de swap)
- (3) *fs_vfstype* – tipul de sistem de fisiere: *ext4, xfs, vfat, ntfs, nfs, proc*, etc (*swap* denota fisierul sau partitia de swap)
- (4) *fs_mntops* – optiunile operatiei *mount* pt sistemul de fisiere respectiv
  - *defaults* inseamna *rw, suid, dev, exec, auto, nouser, async*
  - *user/owner*, permite operatia de *mount* pt utilizator/proprietar
  - *noauto*, mountpoint-ul nu e instalat inb sistem la operatia *mount -a*

Campurile fstab (cont.)

- (5) *fs_req* – folosit pt backup (*dump*), uzual 0
- (6) *fs_passno* – utilizat de *fsck* pt a determina ordinea de verificare la boot
  - radacina / trebuie sa aiba valoarea 1
  - celelalte mountpoint-uri valoarea 2
  - valoare implicita 0 (nu se executa *fsck* la boot), uzual pt remote FS

Obs: *fsck* verifica (si repara daca se poate) un sistem de fisiere

Ex: *$ fsck -t ext4 /dev/sdb2*

$ fsck /dev/sda1

$ fsck /home

De asemenea, *fsck* poate folosi LABELs/UUIDs

Tipuri de sisteme de fisiere

- sisteme de fisiere pt date stocate permanent
  - log-structured/journaling sau nu
- sisteme temporare
- sisteme de fisiere bazate pe echipamente loop
- sisteme de fisiere distribuite
- pseudo-sisteme de fisiere

Log-structured/journaling FS

- actualizarea FS pp multe scrieri
- intreruperi nedorite (system crash, intreruperea curentului) in mijlocul acestor scrieri pot lasa structurile de date interne ale FS-ul intr-o stare inconsistenta
- detectarea acestor inconsistente si recuperarea din starea de inconsistenta pp analiza exhaustiva a structurilor de date FS
  - uzual, implica rularea *fsck*
  - se face inainte ca FS sa fie instalat si folosit din nou pt R/W
  - daca FS are dimensiuni mari si/sau I/O bandwidth-ul e redus => timpi mari de downtime pt sistem
- solutie: *log-structured/journaling FS*
  - aloca un spatiu special (*log/journal*) care stocheaza modificarile care vor fi operate *ulterior* asupra FS
  - dupa crash, se citeste log-ul si se reiau operatiile salvate in log pana cand structurile de date interne FS devin consistente din nou

Log-structured/journaling FS (cont.)

- concret: FS inregistreaza fiecare actualizare a metadatelor ca o tranzactie
- toate tranzactiile sunt scrise intr-un *log/journal*
  - tranzactia e comisa de indata ce a fost scrisa (secvential) in log
  - uneori se foloseste un device separat sau o anumita sectiune a discului
  - in orice caz, in acest moment FS nu poate fi actualizat
- tranzactiile din log sunt scrise *asincron* in structurile de date ale FS
  - cand o asemenea structura s-a modificat, tranzactia e stearsa din log
- la *crash*, toate tranzactiile ramase in log trebuie executate din nou
- consecinte:

(1) recuperarea rapida din crash

(2) indeparteaza posibilitatea aparitiei inconsistentelor in metadate

- exemple: *ext4*, *xfs*, *reiserfs*, samd

Temporary FS (tmpfs)

- sistem de fisiere volatil, stocat in RAM
  - datele stocate in *tmpfs* se pierd dupa dezinstalarea FS (*umount*), dupa reboot sau la caderea curentului electric
- nu este RAM disc, are structura logica de FS, cu abstractii de nivel inalt tip fisiere si directoare
  - RAM disc – disc virtual, organizat dupa principiile HDD
  - eventual, un FS poate rula peste RAM disc
- beneficiaza de spatiul de swap !
  - rezolva problema limitelor de alocare (*out-of-memory*)
- tipul VFS: *tmpfs*

Ex: *$ mount –t tmpfs ramfs /mnt –o size=1g*

- Obs: nu exista device in comanda, inlocuit de un string ales cf dorintei utilizatorului (i.e., *ramfs*)
- utilitate: stocarea datelor temporare (date de configurare severe, FIFO pt IPC programe de sistem, etc), sisteme de fisiere speciale gen *cgroups*, etc

Loop device FS

- prin asocierea unui fisier cu un loop device, continutul fisierului poate fi folosit impreuna cu comanda *mount*
- Ex: CDROM mount

  $ mount -o loop -t iso9660 cdrom.iso /cdrom

- Ex: spatiu de swap care foloseste un fisier in loc de device

  $ dd if=/dev/zero of=~/myswapfile bs=1K count=1M

  $ losetup --show -f ~/myswapfile # pp device-ul ales e /dev/loop0

  $ mkswap /dev/loop0

  $ swapon /dev/loop0

  $ swapon

Loop device FS (cont.)

- Ex: crearea unui sistem de fisiere intr-un fisier

```
$ dd if=/dev/zero of=~/ext4.img bs=1K count=1M
$ losetup --show -f ~/ext4.img        # pp device-ul ales e /dev/loop0
$ mkfs.ext4 /dev/loop0                # creeaza format ext4
$ file ~/ext4.img
$ mount /dev/loop0 /mnt
...
$ umount/dev/loop0
$ losetup -d /dev/loop0
```

Sisteme de fisiere paralele/distribuite

- sisteme message passing, paradigma client-server
- FS server
  - program remote care exporta (o parte a) FS local la distanta
  - uzual se exporta un director de pe masina remote
- FS client
  - program client care serveste ca intermediar intre FS server si VFS-ul local
- Ex: Network File System (NFS)
  - bazat pe Remote Procedure Calls (RPC), executa proceduri la distanta ca si cum ar fi locale
  - procedura locala (i.d., *read*) executata de un *stub client* care contacteaza FS server pt executia procedurii reale la distanta (*stub server*)
  - parametrii de apel si rezultatul operatiei transformati intre formatul de date local si cel de retea (XDR, ASN.1) si inapoi (operatii de *marshalling*/*unmarshalling*)

$ mount –t nfs fmi.unibuc.ro:/home /home

Pseudo-sisteme de fisiere

- in general, interfete catre structurile de date ale kernelului
- *procfs*

  $ mount –t proc proc /proc

- in principal, contine subdirectoare pentru fiecare proces din sistem de forma */proc/[pid]*
- intr-un subdirector *[pid]* se gasesc informatii despre proces, cum ar fi
  - linia de comanda folosita pentru a lansa procesul
  - cpuset-ul procesului
  - cwd, environment, link executabil, file descriptori deschisi (inclusiv 0,1,2)
  - statistici operatii I/O ale procesului
  - acces la paginile de memorie ale procesului + pagemap-ul procesului
  - sistemele de fisiere montate in namespace-ul procesului
  - starea procesului folosita de comanda *ps*
  - informatii despre utilizarea memoriei (dim, dim rezidenta, text, date, lib)
  - descrierea apelului sistem curent (argumente, SP, PC)
  - informatii despre thread-urile si timerele procesului

  samd.

Pseudo-sisteme de fisiere (proc)

- in plus, */proc* contine informatii despre
  - echipamentele PCI (*/proc/bus/pci/devices*, vizibile cu comanda *lspci*)
  - linia de comanda a kernelului la lansare (*/proc/cmdline*)
  - informatii despre CPU si arhitectura lor (*/proc/cpuinfo*)
  - lista nr majore ale echipamentelor (*/proc/devices*)
  - lista sistemelor de fisiere suportate de kernel (*/proc/filesystems*)
  - incarcarea medie a sistemului afisate de comanda *uptime* (*/proc/loadavg*)
  - statistici despre utilizarea memoriei afisate de comanda *free* (*/proc/meminfo*)
  - lista modulelor incarcate in sistem (*/proc/modules*)
  - lista sistemelor de fisiere curent instalate in sistem (*/proc/mounts,* v. *mount*)
  - informatii despre reteaua sistemului, subdirectorul */proc/net*
  - lista partitiilor de disc din sistem (*/proc/partitions*)
  - statisticile kernelului (*/proc/stat*)
  - partitiile de swap in folosinta curenta (*/proc/swaps*)
  - valorile diverselor variabile ale kernelului (*/proc/sys*)
  - statistici despre sistemul de memorie virtuala (*/proc/vmstat*), samd.

Pseudo-sisteme de fisiere (sysfs)

- *sysfs*

$ mount –t sysfs sysfs /sys

- exporta informatii din kernel despre echipamente, module kernel, sisteme de fisiere, etc
- subdirectoare

/sys/block – linkuri simbolice catre */sys/devices* pt fiecare device din sistem

/sys/devices – arborele de structuri de date kernel pt echipamente

/sys/fs – contine subdirectoare pt sisteme de fisiere speciale (e.g., *cgroups*)

/sys/kernel/ - contine fisiere si directoare cu informatii despre starea kernelului

/sys/kernel/mm – contine fisiere si directoare cu informatii despre sistemul de gestiune al memoriei

/sys/module – contine informatii despre fiecare modul kernel din sistem

samd.